UniGuard

Risk management and Insurance mechanisms for Uniswap hooks

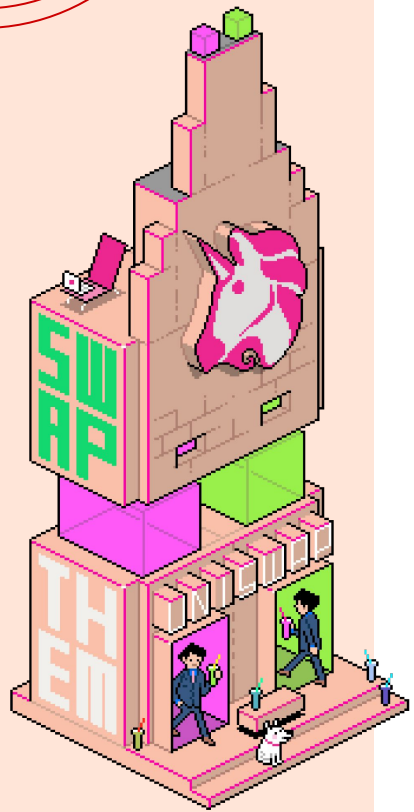
About

An insurance system for Uniswap V4 hooks that combines UNI staking governance, AVS-powered monitoring, and zero-knowledge proofs for victim compensation by Brevis.

Motivation

- Support hook innovation while protecting users
- Create decentralized security validation through UNI staking
- Ensure fair and verifiable compensation for affected users

Triple-Layer Protection System



UNI Staking

Decentralized risk assessment through UNI token staking provides economic security and community governance

AVS Monitoring

Autonomous monitoring system continuously evaluates hook behavior for real-time risk detection

Brevis ZK Proof

Zero-knowledge proof system ensures secure and verifiable victim compensation claims

A comprehensive security framework for Uniswap V4 hooks

Hook Registration Process

Securing Your Hook with Insurance Protection

1. Submit Hook

Developer deploys hook and registers with HookRegistry

Example: LimitOrder.sol

2. Insurance Deposit

Provide USDC deposit to InsuranceVault

Minimum: 10,000 USDC

3. UNI Staking

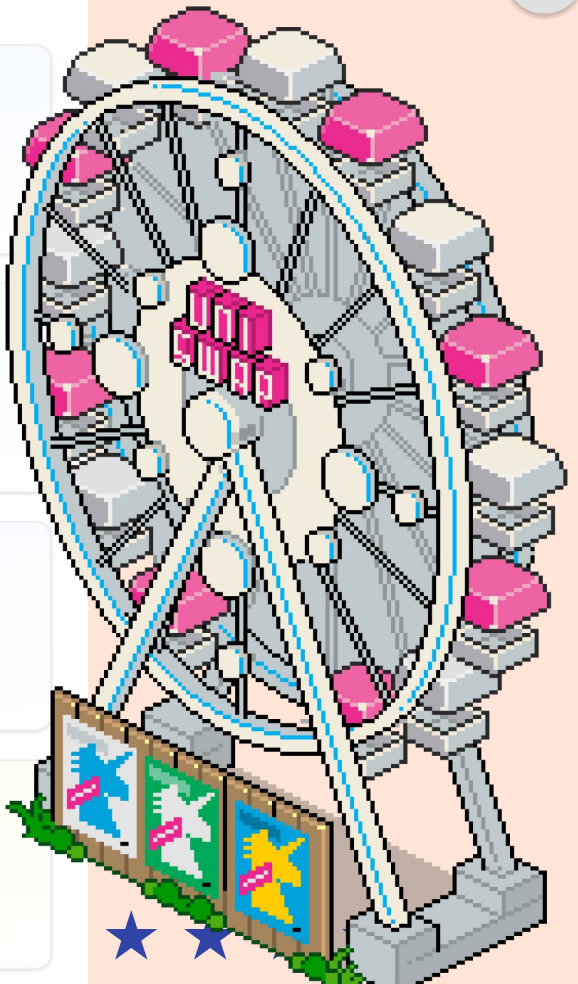
Community stakes UNI tokens to validate hook

Minimum per staker: 1,000 UNI

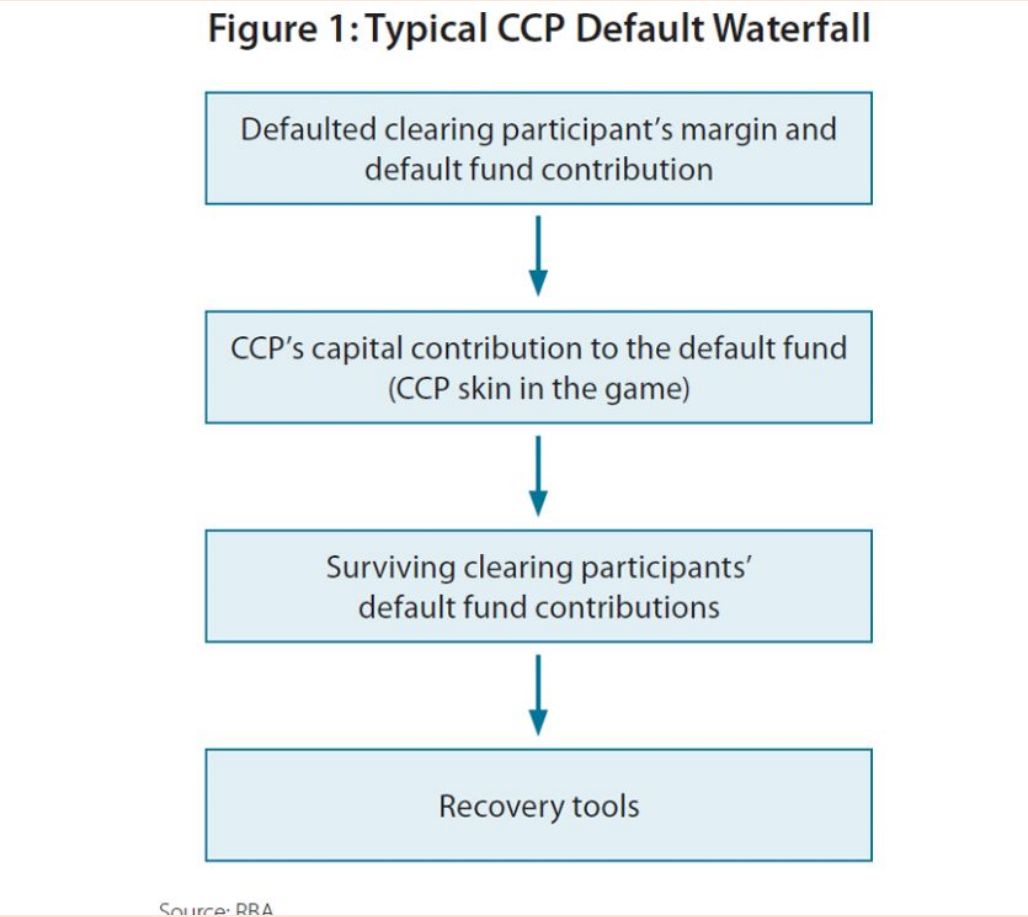
4. Hook Activation

Hook becomes active with full protection

Monitored by AVS + Community



How should we design optimal resolution processes?



In my previous role, I was involved in designing liquidation processes for securities companies in Clearing Corporaion

Want to implement similar resolution mechanisms in the crypto space.

Protection System in Action

Example: Protecting Users from a Compromised Hook

1. Incident Detection

AVS detects suspicious activity in LimitOrder hook:
- Unusual order manipulation detected
- Risk score increases to 80/100

2. Automatic Protection

System responds immediately:
- Hook is automatically paused
- Further interactions blocked

3. Governance Response

UNI stakers take action:
- Insolvency proposal created
- Community votes to process compensation

4. Victim Verification

Brevis processes claims:
- Users submit proof of loss
- ZK proofs verify legitimate claims

5. Compensation

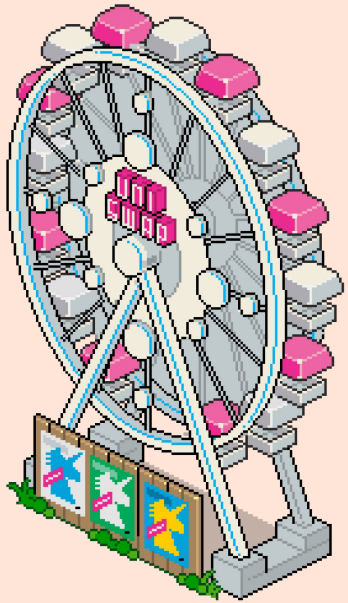
Verified victims receive compensation:
- USDC from insurance pool distributed
- Individual claims processed securely

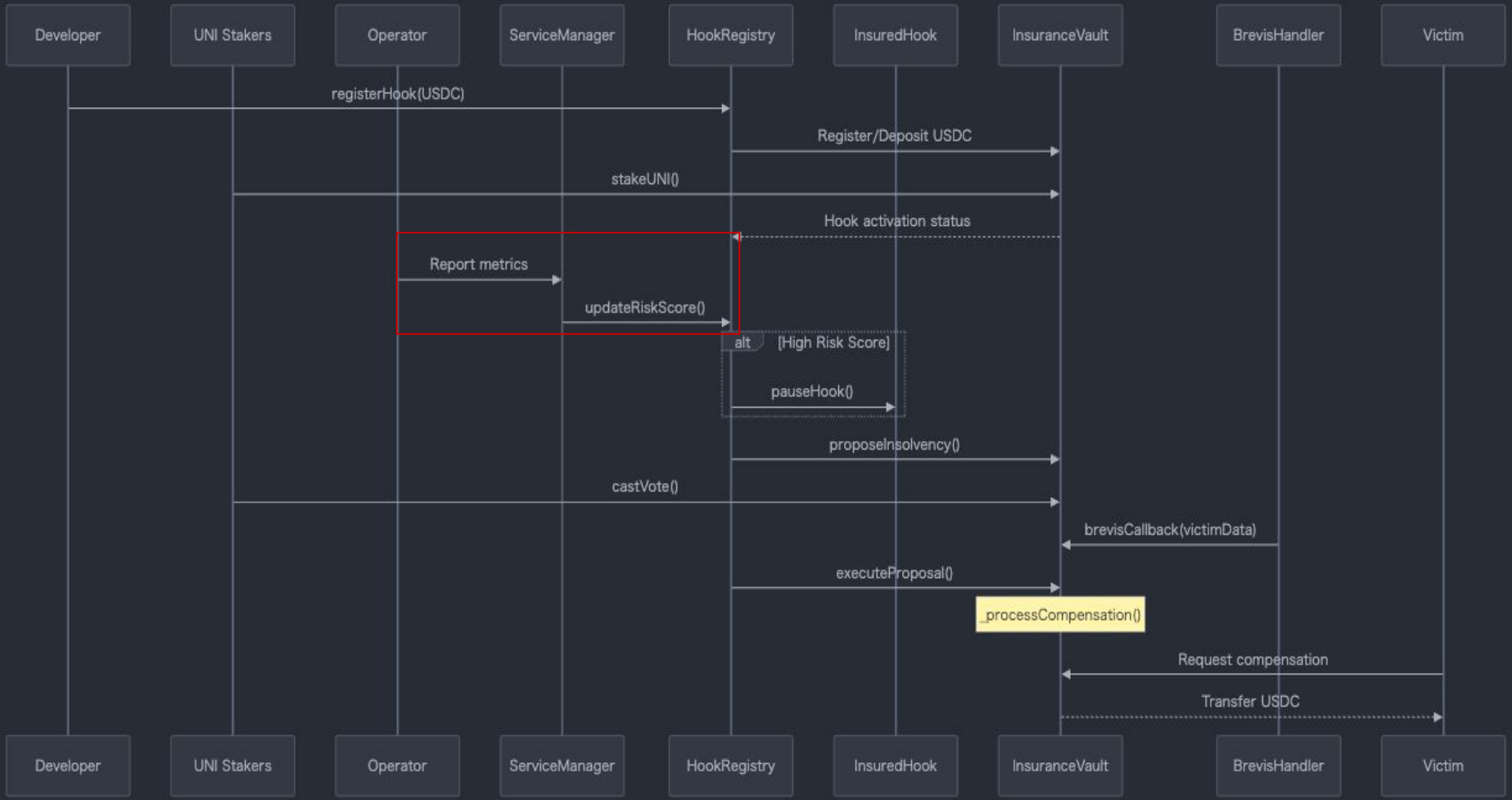
Demo

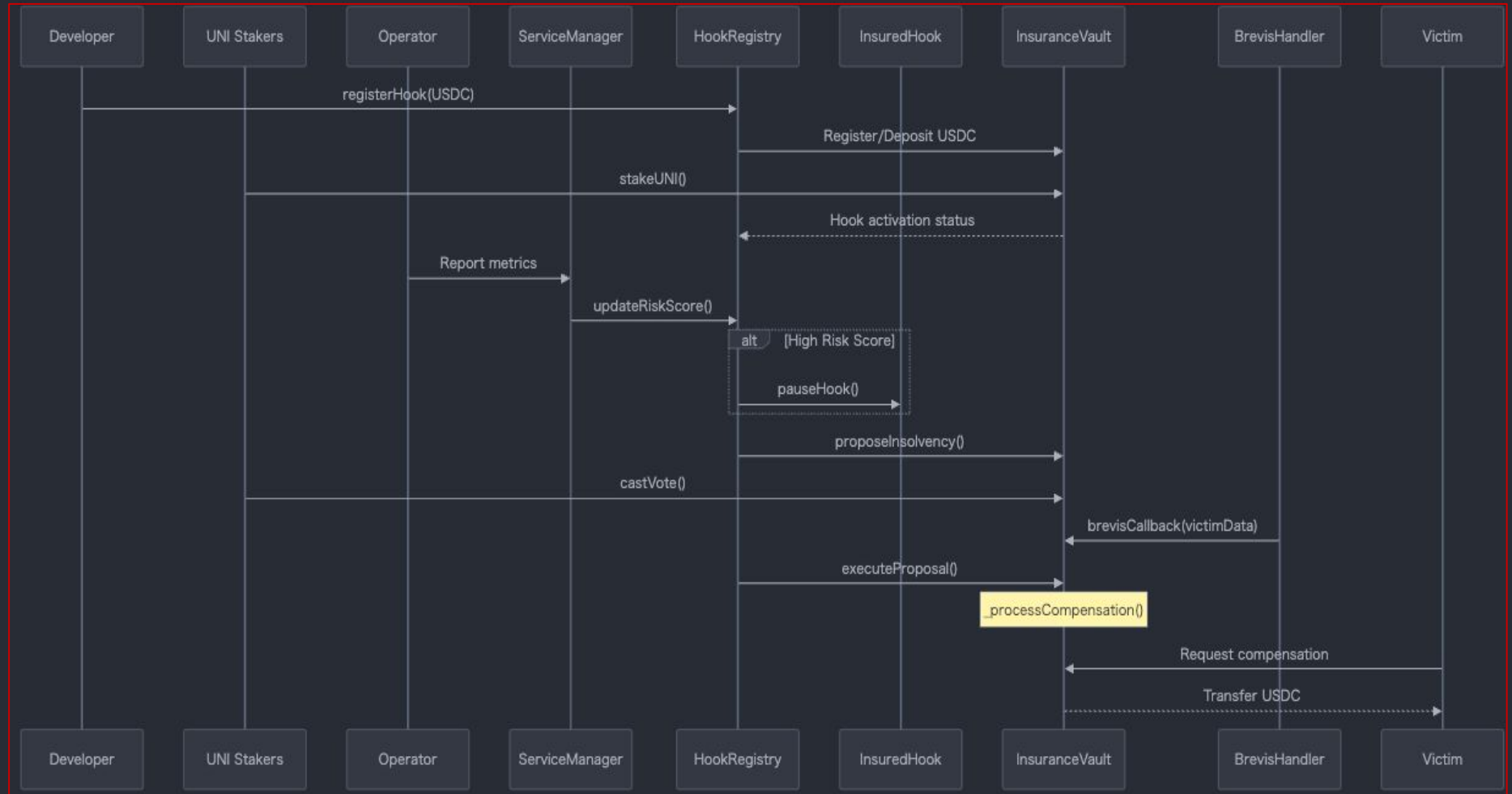
Key scenarios we'll demonstrate:

- Risk monitoring with AVS

- Complete hook lifecycle testing
  1. Regist Hook
  2. Uni token stake
  3. Victim compensation flow
     with Uni governance and Brevis integration




Developer
UNI Stakers
Operator
ServiceManager
HookRegistry
InsuredHook
InsuranceVault
BrevisHandler
Victim
registerHook(USDC)
Register/Deposit USDC
stakeUNI()
Hook activation status
Report metrics
updateRiskScore()
alt
[High Risk Score]
pauseHook()
proposeInsolvency()
castVote()
brevisCallback(victimData)
executeProposal()
_processCompensation()
Request compensation
Transfer USDC
Developer
UNI Stakers
Operator
ServiceManager
HookRegistry
InsuredHook
InsuranceVault
BrevisHandler
Victim


Developer
UNI Stakers
Operator
ServiceManager
HookRegistry
InsuredHook
InsuranceVault
BrevisHandler
Victim
registerHook(USDC)
Register/Deposit USDC
stakeUNI()
Hook activation status
Report metrics
updateRiskScore()
alt
[High Risk Score]
pauseHook()
proposeInsolvency()
castVote()
brevisCallback(victimData)
executeProposal()
_processCompensation()
Request compensation
Transfer USDC
Developer
UNI Stakers
Operator
ServiceManager
HookRegistry
InsuredHook
InsuranceVault
BrevisHandler
Victim

UniGuard is Secure Foundation for Hook Innovation

Combining UNI governance, AVS monitoring, and Brevis ZK proofs

=>Community-driven security validation

This is Uni (:=Uni holder + AVS + Brevis)
Guard